

UNIVERSITY OF SASKATCHEWAN
DEPARTMENT OF COMPUTER SCIENCE

CMPT 250
Final Examination

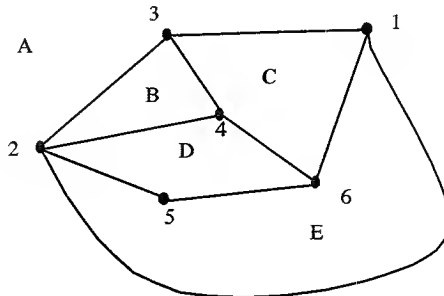
3 hours
Marks

Closed Book

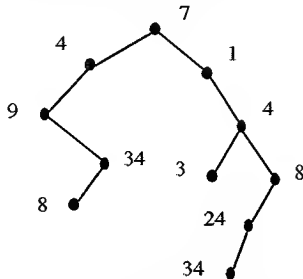
April 22, 2004

This exam is out of 150. Thus, you have 1 minute per mark with 30 minutes to spare. Use this in judging how much time to spend on a question.

- 8 1. Consider the instruction *objf* that might occur in an Eiffel program. When deciding whether this instruction is valid, and deciding how to handle it at runtime, Eiffel uses static-type checking and dynamic binding. Explain the use of static-type checking and of dynamic binding for this instruction.
2. In the language Eiffel, a programmer can use anchoring, i.e., express a type by means of *like v*, where *v* is a variable or function.
- 3 (a) Explain the value of using anchoring
- 2 (b) Specify where anchoring was used to good advantage in the *dslib* data structure library.
3. Consider a B-tree file system.
- 2 (a) A B-tree file system uses a tree type similar to one discussed in detail in class. What is the tree type that was discussed in class?
- 5 (b) Give the general algorithm to access the items of a B-tree file in sequential order.
- 10 4. An undirected graph is called planar if it can be drawn on a flat surface (a plane) in such a way that no two edges cross each other. A planar graph subdivides the plane into regions. The graph below has 5 regions: A, B, C, D and E, where region A is the whole region surrounding the graph. Prove by *mathematical induction* that the number of regions for a connected undirected planar graph is $m - n + 2$, where m is the number of edges, n is the number of vertices, and $m \geq n - 1$.



5. Consider a binary tree that stores integer values. *Do not assume* that the tree is ordered. The objective is to print out the values that occur in *both* the left and right subtrees of the root node. You should assume that no value occurs twice in the left subtree or occurs twice in the right subtree. In the tree below, the values 4, 8, and 34 should be output. Note that the values output can be in any order.



In the parts below, you are to give algorithms and/or code to solve this problem. You need to write the requested routine, but as usual, your routine can call any number of other routines provided that you also write them. In a general algorithm, if a well-known algorithm is called, it is sufficient to refer to the algorithm by name *without* outlining its approach.

You should assume that the tree has type `T` where `T` is a subtype of `LINKED_SIMPLE_TREE_UOS [INTEGER]`. Therefore, the following routines are available for use:

```

make
initialize (lt : like Current; x : INTEGER; rt : like Current)
root_item : INTEGER          out : STRING
is_empty : BOOLEAN          is_full : BOOLEAN
root_left_subtree : like Current  root_right_subtree : like Current
  
```

- 4 (a) Give a general algorithm to print the items in common to the left and right subtrees of the root node. The tree is to be a parameter (formal argument) for the routine. You should not change the tree, and the values are not to be copied or moved into another container.
- 18 (b) Give the Eiffel code for a procedure to implement your general algorithm of part (a).
- 2 (c) Give the order for the time required by your algorithm of part (a)/(b).
- 7 (d) Assume that the tree is very large so that the time requirements are to be minimized. Also, assume that, if appropriate, the values can be copied into another container or containers, and the values can be handled in an appropriate fashion. Give a general algorithm and its time requirements for how to handle the problem with these modified constraints.

- 8 6 Consider the procedure to search for an item in an open-address hash table that uses linear probing. If the item is found, the procedure should set the state of the hash table so that the item being sought is the current item. If the item is not found, the cursor of the hash table should enter the *after* state. In order to ensure that such a procedure is working correctly, what situations should be tested?

7. Consider the following algorithm for an undirected graph:

Main program

```

    For each vertex u
        u.set_reached(false)
    end
    scan( $v_0$ )
end

```

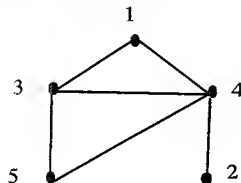
scan(s) algorithm

```

    s.set_reached(true)
    For each w adjacent to s do
        if not w.reached then
            scan(w)
        end
    end
    s.set_reached(false)
end

```

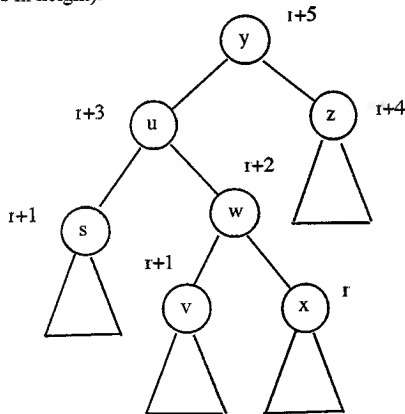
- 2 (a) This algorithm is very similar to an algorithm discussed in class. What is the name of the algorithm discussed in class?
- 2 (b) What is the (very important) difference between this algorithm and the one discussed in class?
- 6 (c) Trace the above algorithm on the graph below, when vertex 1 is used for v_0 . In particular, draw the tree that corresponds to the search, as was done in class for the searches discussed in class. Hence, in the tree include a node for every call of *scan*, label the node with the vertex that is the argument of the call, and add an edge from one call to another call if the first call recursively calls the second call. Note that there will likely be several calls for some vertices.



- 1 (d) If the graph is complete (i.e., it has all possible edges), give the order for the time requirements of the algorithm?

8. Consider height-balanced binary trees.

- (a) Give a (simple) example of insertion into a height-balanced tree that results in a height-balanced tree with height one larger than before the insertion
- (b) Consider the outline of a binary tree given below. A triangle below a node represents the subtree rooted at the node. The height of this subtree is given by the expression beside the node. The letters inside nodes represent the items stored within the nodes, and are used to label the nodes.
- (i) Suppose that because of an insertion, the s subtree has its height **increased** by 1. Does this result in the overall tree becoming out of balance? If so, give as much detail as possible on the rotation that is needed (where, what type)?
- (ii) Repeat part (i) when the v subtree increases in height by 1 (and no other subtree changes in height).
- (iii) Repeat part (i) when the x subtree increases in height by 1 (and no other subtree changes in height).
- (c) Now consider a deletion that reduces a subtree height by 1.
- (i) Suppose that because of a deletion, the s subtree has its height **reduced** by 1. Does this result in the overall tree becoming out of balance? If so, give as much detail as possible on the rotation that is needed (where, what type)?
- (ii) Repeat part (i) when the v subtree reduces in height by 1 (and no other subtree changes in height).
- (iii) Repeat part (i) when the x subtree reduces in height by 1 (and no other subtree changes in height).



9. In class and in the text, the ADT was given for a binary tree. The operations included *make*, *initialize*, *is_empty*, *root_item*, *root_left_subtree*, and *root_right_subtree*. The ADT was defined using the axiomatic approach with *make* and *initialize* as the build operations. Give the axioms to define the following operations for a binary tree:
- insert_root_right(i)*, i an item – returns a tree with the parameter being the root item, the original tree being the left subtree of the root, and an empty right subtree
- rightmost_item* – returns the item of the tree that has no items to its right,
 i.e., its right subtree is empty and it is not in the left subtree of any item

Saskatchewan Government Insurance (SGI) is responsible for the registration and insurance of vehicles in the province. Vehicles are of two main types: motorized and non-motorized. Examples of motorized vehicles are cars and trucks. Non-motorized vehicles are mainly various types of trailers, such as utility trailers, sports travel trailers and industrial trailers.

Cars are categorized as subcompact, compact, mid-size, full-size, van, and SUV. Trucks are categorized as pickup, heavy duty, and semi. Also, for insurance purposes, a vehicle is classified as private, corporate or farm; depending on the vehicle's use (private, business or farming). Corporate owners insure fleets of motorized vehicles and/or industrial trailers. A fleet consists of several vehicles. Corporate owners receive a special discount depending on the size of their fleets. Farm owners also receive a special discount because they are involved in farming (grain, livestock, etc.).

Usually, vehicles are initially registered when purchased new by their original owners. After the initial registration, a vehicle's registration is renewed yearly on the birthdate of its owner. When a vehicle is purchased new by its owner, a record for this vehicle is created. Each vehicle is identified by its serial number. Other attributes of this record include the owner's identification, manufacturer (e.g. Ford, GM, Dodge, Toyota, etc.), year of manufacture, type of vehicle (car, SUV, pickup truck, heavy truck, semi, trailer type, etc.), insurance start date, insurance renewal date, and initial purchase price. The vehicle registration fee can be paid in one yearly payment (the usual case) or over a 12-month period in monthly installments.

The yearly premium for a vehicle depends on the type of vehicle, its cost and the driving record of the owner. An owner with driving infractions, resulting in moving violations and tickets, are charged a higher insurance fee or premium than an owner without any traffic tickets. Also, an owner with no previous accident claims on a vehicle during the last seven (7) years gets a discount on the insurance premium. A driver's license file containing driver license identification with a history of traffic violations and previous vehicle accidents is kept on a separate system.

Compared to a new vehicle, a used vehicle is handled in the same way as a new vehicle, except that a used vehicle is transferred from its previous owner to its new owner. The purchase price is recorded. The previous owner cancels the existing insurance policy on the vehicle, and is reimbursed for part of the insurance premium. The new owner is invoiced for the new insurance premium that he/she pays.

From time to time, an insured vehicle may be deleted from the system because the vehicle is retired, sold to someone out of province, or is destroyed beyond repair in some accident.

Some new vehicles may be purchased by owners in another province such as Alberta, where no provincial tax was charged on the vehicle. Such a new vehicle owner must pay the Saskatchewan provincial tax (PST) on the vehicle when it is registered and insured. This provincial tax is remitted to the provincial government financial system.

The SGI system must also remit the GST and PST tax portions of the insurance premiums to the federal and provincial governments through existing government systems

Vehicle insurance customers receive their insurance renewal notices one month before their birth date. Also, daily, monthly, and yearly reports are required as to insurance premiums collected, GST and PST taxes paid, number and types of vehicles insured, number of insurance claims paid to customers, total value of claims, etc..

For this problem the following are required:

- (a) Give a list of the actors and use cases for the SGI system. For this part, no description is required for any use case – just use a meaningful name.
- (b) Identify and briefly describe the use case for a corporate customer registering and insuring a fleet of new vehicles whose total yearly premium is to be paid in one payment. Also, identify and briefly describe what you consider to be the other two (2) most important use cases in the system.
- (c) Generate an interaction (either a sequence or collaboration) diagram for the use case of registering and insuring a fleet of new vehicles by a corporate customer as described in part (b). If an interaction involves a parameter, be sure to specify the parameter.
- (d) Identify the objects/classes for the SGI system.
- (e) Give a composite (complete) class diagram for the SGI system.
- (f) Describe, using inheritance diagrams with suitable features, the inheritance taxonomies in your design.
- (g) Identify the containers in the system. For each container that is to be stored as a file, give its structure (sequential, direct, or B-tree) and its primary key.

NOTE: You should make use of suitable O-O principles and patterns. The use of these should be identified in your solution. No detailed design is required.

Total 150

The end